

WE CLAIM:

1. A network management system for managing a plurality of network devices,
comprising:
 - a device database for storing a native configuration for one of the plurality of
 - 5 network devices;
 - a component database for storing configuration information used to configure the
plurality of network devices, wherein the configuration information is stored as a
plurality of components and a plurality of candidate components;
 - a device learning module for receiving the native configuration from the device
 - 10 database, identifying the configuration information from the native configuration, and
storing the configuration information in the component database; and
 - a grammar builder for receiving a candidate component, resolving the candidate
component into a component, and storing the component in the component database.
2. The system of claim 1, wherein the configuration information identified from the
- 15 native configuration information comprises a policy.
3. The system of claim 2, wherein the policy comprises a plurality of components.
4. The system of claim 2, wherein the policy comprises a template adapted to aggregate
together a plurality of components all sharing a common attribute.
5. The system of claim 4, wherein the common attribute comprises membership in a
- 20 network configuration function for one of the plurality of network devices.
6. The system of claim 1, wherein the plurality of components comprise a policy-driven
configuration for one of the plurality of network devices.
7. The system of claim 1, wherein each component comprises a component syntax
block.
- 25 8. The system of claim 1, wherein each component comprises a configuration command
for a configuration command language, the configuration command adapted to
configure a function on the network device.
9. The system of claim 1, wherein the configuration information identified from the
native configuration information comprises a candidate component.
- 30 10. The system of claim 1, wherein the grammar builder receives the candidate
component from the device learning module.

11. The system of claim 1, wherein the grammar builder receives the candidate component from the component database.
12. The system of claim 1, wherein the device learning module further comprises a lexer adapted to tokenize the native configuration, and a parser adapted to parse the
5 tokenized native configuration to identify a plurality of components and a candidate component within the tokenized native configuration.
13. The system of claim 12, wherein the parser is configured to identify the plurality of components and the candidate component by comparing the tokenized native configuration to the configuration information stored in the component database.
- 10 14. The system of claim 13, wherein the parser is adapted to identify the candidate component by detecting a portion of the tokenized native configuration that does not match the configuration information stored in the component database.
15. The system of claim 13, wherein the parser is adapted to identify the plurality of components by detecting portions of the tokenized native configuration that match the
15 configuration information stored in the component database.
16. The system of claim 12, wherein the lexer is configured to tokenize the native configuration according to a grammar embodied in the configuration information stored in the component database.
17. The system of claim 12, wherein at least a first portion of the lexer and the parser are
20 recomplied each time the device learning module receives a native configuration.
18. The system of claim 17, wherein a second portion of the lexer and the parser remains constant across multiple receptions of native configurations.
19. The system of claim 12, wherein the device learning module further comprises a policy matcher, adapted to parse the plurality of components and recognize a policy
25 contained in the plurality of components.
20. The system of claim 19, wherein the device learning module is adapted to output a policy-driven configuration and zero or more candidate components.
21. The system of claim 1, wherein the device learning module is adapted to compile a policy-driven configuration into a native configuration.
- 30 22. The system of claim 1, further comprising a device data storage for storing device-specific information about the plurality of network devices.

23. The system of claim 1, wherein the grammar builder is adapted to retrieve command root completion information from one of the plurality of network devices, create a plurality of command root derived components from the command root completion information, and resolve the candidate component into a component by comparing the candidate component to the plurality of command root derived components, to identify one of the plurality of command root derived components that matches the candidate component.

24. A method of parsing a native configuration into a policy-driven configuration, comprising:

receiving configuration information comprising a plurality of components;
receiving the native configuration;
tokenizing the native configuration using a lexer module;
parsing the tokenized native configuration using a parser module, to identify a plurality of input components contained in the native configuration and match the plurality of input components with the plurality of components;
parsing the tokenized native configuration using the parser module, to identify one or more unknown regions contained in the native configuration, which do not match any of the plurality of components;
emitting a tree of components, comprising the plurality of matched input components and the one or more unknown regions;
processing the one or more unknown regions to identify one or more candidate components;
analyzing the tree of components to identify one or more policies present in the tree of components; and
outputting the one or more policies and the one or more candidate components, as a policy-driven configuration.

25. The method of claim 24, further comprising:

compiling the lexer module using the configuration information, such that the lexer module is adapted to tokenize the native configuration according to a grammar embodied in the configuration information;

compiling the parser module using the configuration information, such that the parser module is adapted to match the tokenized native configuration to the configuration information;

26. The method of claim 24, further comprising reorganizing the tree of components to
5 enhance user comprehension of the tree.

27. The method of claim 24, further comprising compressing the tree to remove unnecessary data from the tree.

28. The method of claim 27, wherein the unnecessary data comprises an empty node in the tree.

10 29. The method of claim 24, wherein the plurality of components comprises a policy, and analyzing the tree of components comprises:

retrieving the policy;

comparing each of the plurality of components in the policy with the plurality of matched input components contained in the tree of components;

15 aborting the analysis if any of the plurality of components in the policy is not found in the plurality of matched input components; and

identifying the policy as present in the tree of components if all of the plurality of components in the policy are found in the plurality of matched input components.

20 30. The method of claim 29, further comprising removing all of the components in the plurality of matched input components that were matched to the plurality of components in the policy, and inserting the policy into the tree of components.

31. The method of claim 29, wherein the policy comprises an ordered policy, and identifying the policy further comprises identifying the ordered policy as present in the tree of components if all of the plurality of components in the ordered policy are
25 found in the plurality of matched input components in the same order as in the ordered policy.

32. The method of claim 24, wherein parsing the tokenized native configuration to identify one or more unknown regions comprises:

30 detecting a parse error, wherein the parser fails to recognize a portion of the tokenized native configuration;

marking the portion for further processing; and

continuing to parse the tokenized native configuration.

33. The method of claim 24, wherein processing the one or more unknown regions to identify one or more candidate components comprises:

walking the tree of components;

5 marking each of the plurality of matched input components in the tree of components;

re-walking the tree of components;

identifying a token within one of the one or more unknown regions within the tree of components.

10 walking the tree of components in each of two directions, beginning at the identified token, to identify a beginning point and an end point of the unknown region; and

marking the unknown region as a candidate component.

15